

WHAT IS CLAIMED IS:

1. A method for converting original computer source code (100) to target source code (500) comprising the steps of,

resolving said original computer source code into a plurality of basic constituent elements (3000) and a plurality of interrelationships therebetween,

storing said elements in a database (30) which retains said interrelationships between said elements, and

writing said target source code from at least one template incorporating said elements therein, said template defining the structure of said target source code and comprising an algorithmic template language for controlling said writing.
2. The method of claim 1 further comprising the steps of,

including a condition statement (3147) in said at least one template, said condition statement having a condition expression contained therein,

evaluating said condition expression,

performing a first task if said condition expression is evaluated to be a first value, and

performing a second task if said condition expression is evaluated to be a second value.
3. The method of claim 1 further comprising the steps of,

including an iteration statement (3148) in said at least one template, said iteration statement having an iteration expression contained therein,

iteratively evaluating said iteration expression, and

performing a task while said iteration expression is evaluated to be a particular value.
4. The method of claim 1 further comprising the steps of,

including a subroutine statement (3149) in said template, said subroutine statement having a subroutine name contained therein,

designating a portion of said at least one template by said subroutine name, and interpreting said subroutine statement as a directive to control said writing by said portion.

5. The method of claim 4 further comprising the step of, when said portion has completed controlling said writing, controlling said writing by said template language immediately following said subroutine statement.

6. The method of claim 1 further comprising the steps of, including a set statement in said at least one template, said set statement having a set expression and a set variable contained therein, assigning said set expression to said set variable.

7. A system for converting original computer source code (100) representing a business logic to target source code (500) comprising,

at least one computer system having at least one template (314), a deconstruction program (20), a database (30) operably coupled with said deconstruction program, and a construction program (40) operably coupled with said database,

said original source code forming an input to said deconstruction program resulting in an output of said deconstruction program of a plurality of basic elements (3000) and a plurality of interrelationships therebetween, said output of said deconstruction program retaining said business logic and stored in said database,

said template characterized by an algorithmic template language included therein which controls said construction program and which generally defines the structure of said target source code,

said construction program arranged and designed to receive said plurality of basic elements and said plurality of interrelationships therebetween stored in said database as a first input, receive said at least one template as a second input, and produce as an output said target source code as an output.

8. The system of claim 7 wherein,
said deconstruction program includes a language-determination parser.
9. The system of claim 7 wherein,
said deconstruction program includes at least one language-dependent parser.
10. The system of claim 7 wherein,
said database stores said original source code.
11. The system of claim 7 wherein,
said database stores an original environment description.
12. The system of claim 7 wherein,
said database stores a target environment description.
13. The system of claim 7 wherein,
said database stores at least one rule for controlling said construction program.
14. The system of claim 7 wherein,
said database stores at least one core construct.
15. The system of claim 7 wherein,
said database stores at least one template.

16. The system of claim 7 wherein,
said construction program includes a template workbench designed and arranged for editing said at least one template.
17. The system of claim 13 wherein,
said construction program includes a rules workbench designed and arranged for editing said at least one rule.
18. The system of claim 7 wherein,
said template language includes a condition statement (3147) having a condition expression contained therein,
said construction program being designed and arranged to interpret said condition statement as a directive to evaluate said condition expression and perform a first task if said condition expression is evaluated to be a first value and perform a second task if said condition expression is evaluated to be a second value.
19. The system of claim 7 wherein,
said template language includes an iteration statement (3148) having an iteration expression contained therein,
said construction program being designed and arranged to interpret said iteration statement as a directive to iteratively evaluate said iteration expression and perform a task while said iteration expression is evaluated to be a particular value.
20. The system of claim 7 wherein,
said template language includes a subroutine statement (3149) having a subroutine name contained therein,

said at least one template (314) having a portion therein designated by said subroutine name,

said construction program being designed and arranged to interpret said subroutine statement as a directive to interpret said portion.

21. The system of claim 20 wherein,

when said construction program has completed interpreting said portion, said construction program interprets said template language immediately following said subroutine statement.

22. The system of claim 7 wherein,

said template language includes a set statement having a set expression and a set variable contained therein,

said construction program being designed and arranged to interpret said set statement as a directive to assign said set expression to said set variable.

23. A system for converting original computer source code (100) to target source code (500) comprising,

at least one computer system having a deconstruction means (20), a database (30), at least one template (314), and a construction means (40),

said deconstruction means being arranged and designed to resolve said original source code into a plurality of basic constituent elements (3000) and a plurality of interrelationships therebetween,

said database adapted for storing said elements and interrelationships,

said template defining generally the structure of said target source code and comprising an algorithmic template language (3144) designed and arranged for controlling said construction means, wherein

said construction means is arranged and designed to interpret said template and write said target source code therefrom, incorporating said elements and interrelationships as directed by said template.

24. The method of claim 23 wherein,

said template language includes a condition evaluation means to control said writing in a first manner if said condition expression is evaluated to be a first value and control said writing in a second manner if said condition expression is evaluated to be a second value.

25. The method of claim 23 wherein,

said template language includes a looping means to iteratively evaluate a looping expression and iteratively control said writing in a predetermined manner while said iteration expression is evaluated to be a particular value.

26. The method of claim 23 wherein,

said template language includes a means to include subroutines.

27. The method of claim 23 wherein,

said template language includes a means to assign a variable a predetermined value.